1948-06-10

A Theory of Self-Checking and Self-Correcting Codes -- Case 20878

Hamming, Richard W.

Monterey, California: Naval Postgraduate School

E-1932-B (7-47)

BELL TELEPHONE LABORATORIES
INCORPORATED

COVER SHEET FOR TECHNICAL MEMORANDA

SUBJECT: A Theory of Self-Checking and Self-Correcting Codes - Case 20878

COPIES TO:

1 — R.L.D.-H.W.B.-H.F.-Case Files
2 — Case Files
3 — T. C. Fry
4 — B. D. Holbrook
5 — E. G. Andrews
6 — C. E. Shannon
7 — J. Riordan
8-- B. McMillan
9-- S. A. Schelkunoff
10-- Dept. 1000 Files

MM— 48-110-31
DATE June 10, 1948
AUTHOR R. W. Hamming
Filing Subject Signalling Systems

ABSTRACT

The two-out-of-five-hole codes are examples of codes which send extra information to be used to detect any single error. By sending even more extra information it is possible to devise methods for locating and correcting errors. The problem examined in this memorandum is that of constructing typical self-checking and self-correcting codes in which the amount of extra information sent is kept reasonably small.

A Theory of Self-Checking and Self-Correcting Codes - Case 20878

MM-48-110-31

June 10, 1948

MEMORANDUM FOR FILE

1. Introduction

A self-checking code is a code in which one or more errors may be detected. For example, in the two-out-of-five-hole codes used in the Laboratories, a single error will cause an odd number of holes to appear, and this is easily detected. Some multiple errors may also be detected. This memorandum introduces a simple basic type of self-checking code, called an "n + 1 code," which includes the two-out-of-five-hole codes as special cases. The n + 1 codes are "best" in the sense that no other self-checking code can send more information with the same number of symbols in the same basic groups that are being checked.

Self-correcting codes are a natural development of self-checking codes. Self-checking codes merely detect the presence of an error or errors; self-correcting codes also give the information necessary to correct these errors. The number of independent errors which may be corrected depends on the code used.

The price of the self-correcting feature, like that of self-checking, is in the requirement that more than the minimum number of symbols must be sent, as well as in the extra equipment for encoding, locating, and correcting the errors that occur. This memorandum concerns itself mainly with the first aspect, which is measured by the redundancy, that is, the ratio of the number of symbols sent to the minimum number necessary to convey the same information without any checks.

One of the most likely applications of self-correcting codes is to high speed digital machines, and it was in this connection that they were developed. The Relay Computer, operating with a self-checking code, stops whenever an error is detected. Under normal operating conditions this amounts to two or three stops per day. However, if we imagine a comparable electronic computing machine operating at 10^5 times the speed and with elements 10^3 times more reliable than relays, we find two to three hundred stops per day. A self-correcting code, with suitable equipment to effect the corrections, would reduce the stops per day to the vanishing point. In fact, a self-correcting code would enable one to relax the reliability

requirements on the individual elements and still have the stops per day vanishingly small. It is hoped that the amount of equipment required to mechanize a self-correcting code would not be excessive, but this question requires further examination.

This memorandum is not a complete study, but is rather an introduction to the field. As such it gives some of the basic concepts involved, and limits the study to a few simple cases. A later memorandum will examine some codes having a minimum redundancy.

Because of the wide use of the two-out-of-five-hole codes in the Laboratories, a brief examination has been made of some of the possible modifications of these codes to convert them into self-correcting codes.

The basic methods were developed in joint discussions between C. E. Shannon, B. McMillan, and R. W. Hamming.

2. The n + 1 Codes

Self-checking codes are not new. The device of sending a word count at the end of a message is a simple example of a partially self-checking code. The two-out-of-five-hole codes are a better example since they detect any one error. The n + 1 codes we are about to describe are the simplest self-checking codes as well as the most efficient in a mathematical sense.

We shall suppose we are sending information in the binary code. Typical examples of systems which are equivalent to the binary code are open and closed relays, and flip-flop circuits. For purposes of notation these two states of equilibrium will be labeled 0 and 1. If the apparatus used admitted of k states ($k > 2$), then k meaningful symbols 0, 1, 2, ..., k-1 could be used to transmit information. This would require only minor modifications in the subsequent analysis which is based on the use of the binary system.

We break up the message into groups of n consecutive symbols, and at the end of each group add a single check symbol which will be chosen so that there will be an even number of 1's in the total n + 1 symbols. From a practical point of view it is advantageous to regard the first n symbols as the message and the last one as the check, but from a theoretical point of view it is usually better to look at the n + 1 symbols as being the message with each symbol playing an equal role to any other symbol. In this code any single error can be detected.

Two situations require examination in attempting to detect an error, depending on whether the digits are presented at the same time (in parallel) or at different times (in

sequence). In the first case we imagine a relay being operated for each 1 symbol. The relays are wired so that each operated relay transposes two wires, while each non-operated relay leaves them the same. The excess or deficiency of a single 1 symbol is thus revealed by the net result that the two wires are transposed. In the second case we imagine the 1 symbols actuating the conventional W-Z combination of relays.[1] Unless the W-Z circuit returns to its original state after the n + 1 symbols have passed there is an error in the group.

Let us now compare the n + 1 code with the two-out-of-five-hole codes. In the n + 1 code the first four binary digits may be used to send any one of 15 numbers (we are excluding the zero symbol),[2] while the fifth binary digit provides the check. Of these 15 groups of five binary digits, 5 have four 1's and 10 have two 1's. The two-out-of-five-hole codes make use of only the 10 combinations involving two 1's, while discarding those with four 1's.

3. Simple Error-Correcting Codes

If an error can be not only detected but its position located as well, then it can be corrected. As the simplest example of locating an error as well as detecting it, let us imagine a message of length $M = n^2$ with its elements arranged in the form of a square of side n. We now bound the square on two adjacent sides with 2n + 1 check symbols. Each check symbol, except the corner one, is at the end of a row or a column, and each is set to make the number of 1's in the row or column an even number. That the corner symbol can be set to check both its row and column depends on the fact that it makes the number of 1's in the entire message an even number. We now pass the symbols through checking equipment, and if a row and a column fail to check that indicates that the element in that row and column is in error (provided we know that at most a single error has occurred). Thus we can locate any single error whether it be in the message or in the checking symbols--all are checked equally.

If two errors occur their presence can be detected, but their positions cannot be located uniquely by this code, since if two rows and two columns fail to check there are four

---

[1] This is the relay analogue of a flip-flop circuit in electronics.

[2] If we wished to use all 16 combinations and still avoid having a symbol with all 0's, we could change the rule for checking and set the check symbol so that an odd number of 1's appeared in the absence of an error.

possible positions for the errors and we cannot tell which diagonal two are wrong, while if the two errors are on the same row (or column) only two columns (or rows) will fail to check and these have no common members.

It is perhaps worth noting that the corner symbol need not be included. If it is not used, then when both a row and a column fail to check we still get the coordinates of the erroneous symbol, while if a single row or column fails to check we know that the check symbol itself is in error. However, two errors are no longer detectable without the corner check symbol.

The central concept of this method of locating errors is that of dividing the entire message up into sets such that each symbol is the unique common member of two sets. In the example above we used rows and columns. In particular cases it may be preferable to use, say, rows and diagonals in order to simplify special equipment.

In the above example we used a square, though clearly any rectangle will do. In general one can say that the square is preferable since the checking symbols are a semi-perimeter, and for a square this is less than the semi-perimeter of any rectangle having the same area. Another way of putting this same point is to say that for a square all the checks are working equally hard, that is, all are checking the same amounts of information. For this reason, and to keep the study in bounds, we shall restrict ourselves mainly to the study of square codes.

This simple code may be viewed in another light; it may be regarded as a double n + 1 code. Each of the rows may be looked upon as a simple n + 1 code which is capable of detecting but not locating an error. We now look upon these blocks of n + 1 symbols as our basic blocks in sending information. Out of these blocks we construct an n + 1 code, using the n + 1 block as a checking block. We have only to define what we mean by a checking block, and one suitable definition is the one that we have used regarding the even number of 1's in each column (position in the block). This new check provides the necessary cross checks which enable us to locate the error.

4. A Generalization

The two preceding codes have been generalized in a number of ways; of these we shall give only one.

The next code in our particular induction is obtained by going to three dimensions and forming a triple n + 1 code in the form of a cube. Upon examination it will be found that any two or three errors can be located with no ambiguity, and that many combinations of more errors can be located as well. The most likely combination of errors which is not locatable, but

is detectable, is four errors arranged in the form of a rectangle lying in a plane parallel to one of the faces of the cube. The combination which is most likely to be misinterpreted is five errors, four of which are arranged as above and the fifth in line with one of the corners.

In continuing this particular induction we pass on to higher dimensional codes (the actual symbols, of course, need not be supposed to be arranged in other than a line if desired), being able to correct more and more independent errors. This is not profitable for two reasons:

a. the redundancy approaches infinity, and

b. as we shall show shortly in most situations there is no need at present for correcting more than single errors.

5. Evaluation of the Codes

When a message in one of the preceding codes is received one of three situations occurs:

A. The message is correct or is correctable.

B. The message has errors which are detectable but are not correctable.

C. The message has errors which are either not detectable, or if detectable will be corrected wrong since they appear to be caused by another combination of errors.

If we measure the probability of occurrence of the classes A, B, and C by the numbers A, B, and C, then

$$A + B + C = 1 .$$

These three numbers, together with the redundancy R, serve to measure the theoretical properties of a code. In practice, of course, the cost of the actual mechanization is also of great importance. To simplify the mechanization we might agree to detect but leave uncorrected various kinds of theoretically correctable errors; this would clearly affect the realized values of A, B, and C.

In order to calculate the numbers A, B, and C we shall assume that any single error occurs with probability p and is independent of any other error. This assumption of the independence of the errors is not in fact a true one, but when not pushed too far is a reasonable approximation to the true situation.

We first calculate the quantities for an uncoded message of length M. Set $q = 1 - p$. We find

$$A = q^M = (1 - p)^M$$

$$B = 0$$

$$C = 1 - q^M$$

$$R = 1 \quad .$$

We next calculate the quantities for the n + 1 codes, setting the coded message length $N = M + 1$. We have immediately that

$$A = q^N = (1 - p)^N \ .$$

The value of C is the sum of the probabilities of two errors, four errors, six errors, etc., that is,

$$C = C_{N,2}p^2q^{N-2} + C_{N,4}p^4q^{N-4} + C_{N,6}p^6q^{N-6} + \cdots$$

$$= \frac{1}{2} + \frac{1}{2}(q - p)^N - q^N \quad .$$

From this we find

$$B = 1 - A - C$$

$$= \frac{1}{2} - \frac{1}{2}(q - p)^N \ .$$

Finally,

$$R = \frac{n + 1}{n} = 1 + \frac{1}{M} \ .$$

We now turn to the square codes where $M = n^2$ and $N = (n + 1)^2$. Since we can correct a single error

$$A = q^N + C_{N,1}pq^{N-1} = q^{N-1}[1 + (N - 1)p] \ .$$

In calculating C it is not feasible to examine all possible situations so we shall restrict our attention to those which involve not more than four errors. The major sources of errors are three errors which are located on the corners of a rectangle since in checking they appear as if the one remaining corner were in error, and four errors which occur on the four corners of a rectangle since then no error is detected,

$$C = Nn^2p^3q^{N-3} + \frac{1}{4} Nn^2p^4q^{N-4}$$

$$= MNp^3q^{N-4}\left[1 - \frac{3}{4} p\right] .$$

From this we find

$$B = 1 - A - C .$$

Finally

$$R = \left(\frac{n + 1}{n}\right)^2 = 1 + \frac{2}{\sqrt{M}} + \frac{1}{M} .$$

Lastly, for the cubic codes we find that if we try to correct only combinations of three errors (the rest are probably too expensive to mechanize for what they contribute), and consider at most those involving five errors, then with $N = (n + 1)^3$ and $M = n^3$,

$$A = q^N + C_{N,1}pq^{N-1} + C_{N,2}p^2q^{N-2} + C_{N,3}p^3q^{N-3}$$

$$= q^{N-3}\left[1 + (N - 3)p + \frac{N^2 - 5N + 6}{2} p^2 + \frac{N^3 - 6N^2 + 11N - 6}{6} p^3\right]$$

$$C = 3MNp^5q^{N-5}$$

$$B = 1 - A - C$$

$$R = \left(\frac{n + 1}{n}\right)^3 = 1 + \frac{3}{\sqrt[3]{M}} + \frac{3}{\sqrt[3]{M^2}} + \frac{1}{M} .$$

We now consider what values of p may be expected in practice. In some kinds of toll signalling p may be as high as 10^{-3} to 10^{-4}, while for inside plant equipment it is more like 10^{-6} to 10^{-9}. For many kinds of electronic equipment p may be in the range 10^{-9} to 10^{-12}.

In cases where the message length is definitely less than p^{-1}, and this is usually the situation outside of accounting practice, we may neglect terms in p^{k+1} in comparison with terms in p^{k}. The results we have calculated so far may then be summarized in the following table:

Table 1

Code	A	B	C	R = N/M where
0	$1 - Mp$	0	Mp	$N = M$
1	$1 - Np$	Np	$\frac{N(N-1)}{2} p^2$	$N = M + 1$
2	$1 - \frac{N(N-1)}{2} p^2$	$\frac{N(N-1)}{2} p^2$	MNp^3	$N = (\sqrt{M} + 1)^2$
3	$1 - C_{N,4}p^4$	$C_{N,4}p^4$	$3MNp^5$	$N = (\sqrt[3]{M} + 1)^3$

where M is the message length and N the number of symbols actually sent.

On the other hand, in case $M = p^{-1}$ we find

Table 2

Code	A	B	C	R
0	.368	0	.632	1
1	.368	.442	.190	$1 + p$
2	.736	.264	p	$1 + 2\sqrt{p}$
3	.983	.017	$3p^3$	$1 + 3\sqrt[3]{p}$

If we plot the redundancy R as a function of the message length M we find the curves given in Fig. 1. These curves have meaning only at certain points, but have been drawn as smooth curves to give the general behavior in as simple a manner as possible. Table 2 indicates their behavior for large M.

In examining the other columns of Table 1 we note first that insofar as we want to get the correct answer (Column A) an uncoded message is better than an n + 1 code; it is in avoiding errors (Column C) that an n + 1 code is useful. Usually we want both to avoid errors and to increase our chances of getting the correct answer, and for this we can turn to codes 2 and 3.

Numerical examples show that in most cases code 3 is not realistic. For example, if we take $p = 10^{-6}$ and $M = 64$ we find $B = 10^{-17}$ and $C = 2.4 \times 10^{-26}$. These are so small that we immediately suspect that our approximation of the independence of errors is not justified.

This argument indicates that the single error correcting codes, in particular code 2, are the only realistic methods of increasing our chances of getting the correct answer in an economical manner while still using equipment having the same reliability of the individual components. We also note that any improvement in p is reflected in relatively greater improvements in code 2.

6. One Application to Two-Out-of-Five-Hole Codes

In this section we shall examine one possible application of the foregoing codes to the usual two-out-of-five-hole codes used by the Laboratories.

What we wish to examine is not the absolute redundancy, but the extra redundancy in going from the particular self-checking code to a self-correcting code. This we can do by studying the relative redundancy $R_b(A)$ of code A over code B which we define as

$$R_b(A) = \frac{R(A)}{R(B)} .$$

The simplest way of forming a self-correcting code from the usual two-out-of-five-hole code is to form an n + 1 code by sending first n two-out-of-five-hole blocks followed by a single check block. This check block will supply the cross checks necessary to locate any isolated error, since the code is already self-checking.

In choosing the value of the check block there are several ways of proceeding. If we imagine our basic blocks arranged one above the other in the form of a rectangle, then the usual rule we have used would fix the positions in the check block so that an even number of holes appeared in each column. This leads to a check block which may have 0, 2, or 4 holes. But a check block with no holes is highly undesirable since there will be nothing to indicate it except blank paper. One method of avoiding a check block with no holes is to change our rule for computing it so that each column has an odd number of holes; this produces check blocks with 1, 3, or 5 holes only.

In order to calculate the quantities A, B, and C which help measure the code it is necessary to introduce two probabilities of errors; p_0 as the probability that a hole will fail to be punched, and p_1 as the probability that a hole will be punched where there should be a blank. From these we can calculate

Q = probability that all 5 positions are correct,

P = probability of an error which is detectable in the basic block of five,

U = probability of an error that is not detectable in the basic block of five,

where, of course,

$$Q + P + U = 1 .$$

Since Q and P are easier to measure than p_0 and p_1, it seems best to base our calculations on Q, P, and U. We assume P very small, and $P^2 \sim U$.

We have for the original n blocks of the error-detecting code

$$A_{ED} \sim Q^n \sim 1 - nP$$

$$B_{ED} \sim 1 - (A + C) \sim nP$$

$$C_{ED} \sim nUQ^{n-1} \sim nU \quad .$$

For the proposed n + 1 error correcting code we have

$$A_{EC} \sim Q^{n+1} + (n+1)PQ^n$$

$$B_{EC} \sim \frac{(n+1)n}{2} P^2 Q^{n-1} + (n+1)UQ^n$$

$$C_{EC} \sim UP$$

Thus we find for the relative values of the error-correcting code over the error-detecting code

$$A = \frac{A_{EC}}{A_{ED}} \sim 1 + nP$$

$$B = \frac{B_{EC}}{B_{ED}} \sim nP$$

$$C = \frac{C_{EC}}{C_{ED}} \sim \frac{P}{n}$$

$$R_D(C) = 1 + \frac{1}{n} \quad .$$

From this we see that for $\frac{1}{n}$ extra message we have decreased by a factor of order P both the chance of finding an uncorrectable error and the chance of committing an error.

R. W. HAMMING

Att.
BA-397593

ISSUE	DR.	TITLE	BELL TELEPHONE LABORATORIES, INC.
June 9, 1948	T.T.A		NO. OF SHEETS PER SET SEE SHEET 1
	ENG. R.W.H.		BA-397593 SHEET

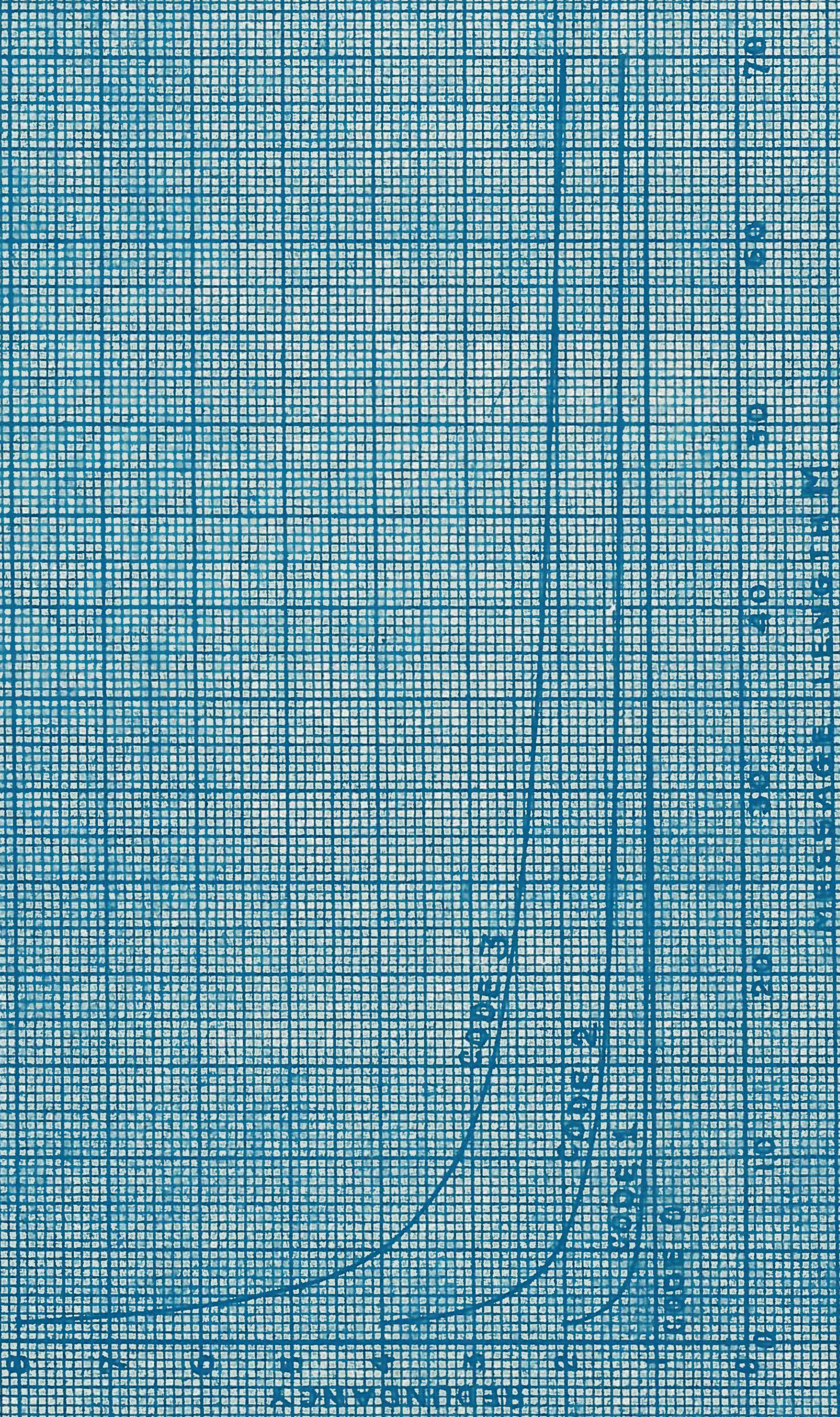

FIG. 1